1.

a. O(n) increases by 1 when n is in the series.

2, 2^2, 2^2^2, 2^2^2^2, etc.
O(2) n=16
O(3) n=256
O(4) n=65536

O(x) n=2^2^(x)
Solve for n O(log(log(n)))

Final = θ(log(log(n)))

b. "for(int i=0; i<=n; i++)" runs n times
Only triggers inner loop sqrt(n) times.
an O(1) event gets triggered i^3 times

n=36: 6^3 + 12^3 + 18^3
n=49: 7^3 + 14^3 + 21^3

From o->sqrt(n) add i^3

sqrt(n)^3sqrt(n)^4 = O(sqrt(n)* n^3)

Final = θ(sqrt(n)*n^3)

c. First loop runs in O(n) time
inside loop runs only approximately log(n) times
triggers another O(n) loop
input 100 output ~= 300
input 1000 output ~= 3000

Final θ(n(log(n)))

d. call rfunc

```
        2 things that do something

        loop(N,n==sqrt(n))
                O(1)
                O(1)
                Rhelp(n)
                Rfunc(n-m, m)
```

O(n/sqrt(n)*O(2)+O(n))

Final = θ(n^2/sqrt(n))